Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

Keras Model for Text Classification in Amazon Review Dataset using LSTM

[1]Thivaharan .S, [2]Srivatsun .G,

[1]Asst.Prof (Sel. Grade), CSE,
PSG Institute of Technology and Applied Research,
Coimbatore, India.

[2]Associate Prof, ECE,
PSG College of Technology,
Coimbatore, India.

Abstract- With the use of Ecommerce, Industry 4.0 is being effectively used in online product-based commercial transactions. An effort has been made in this article to extract positive and negative sentiments from Amazon review datasets. This will give an upper hold to the purchaser to decide upon a particular product, without considering the manual rating given in the reviews. Even the number words in an inherent positive review exceeds by one, where the present classifiers misclassify them under negative category. This article addresses the aforementioned issue by using LSTM (Long-Short-Term-Memory) model, as LSTM model has a feedback mechanism based progression unlike the other classifiers, which are dependent on feed-forward mechanism. For achieving better classification accuracy, the dataset is initially processed and a total of 100239 short and 411313 long reviews have been obtained. With the appropriate Epoch iterations, it is observed that, this proposed model has gain the ability to classify with 89% accuracy, while maintaining a non-bias between the train and test datasets. The entire model is deployed in TensorFlow2.1.0 platform by using the Keras framework and python 3.6.0.

Keywords: Ecommerce, LSTM model, feed-forward mechanism, feedback mechanism, Keras, TensorFlow, relu, sigmoid, GPU device.

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

1. INTRODUCTION

1.1 PROBLEM STATEMENT

This article implements the LSTM based text classification over the Amazon Review dataset. The execution platform used here is TensorFlow 2.1 [1]. TensorFlow is an open source platform, which has proven results while working with machine learning in an end-to-end interactive mode. The presently available sentiment analyzers [2] are not reliable in all the cases, as they were deployed for a particular scenario. The Amazon review dataset is used (2018), while considering the cell phone and new accessories related review over a small subset of the dataset, which is taken in to consideration. The LSTM model is used when compared to other classifier because of the feedback mechanism [3] unlike the feed forward mechanism [4]. The dataset has initially undergone usual processing. After the initial processing, LSTM model along with over-fitting restrictors like “ReLU” and “sigmoid” [5] and embedding model with 16-dimension is created. The actual expected result is out from the dataset evaluation for classification by considering the reviews, which have rating 3 and above (i.e. 3, 4 and 5) as positive sentiments and reviews which have rating below 3 (i.e. 1 and 2) as negative sentiments.

1.2 MODEL INITIALIZATION FOR CLASSIFICATION

The following libraries need to be imported for the purpose of classification: matplotlib, matplotlib.pyplot, os, re, shutil, string, tensorflow, tensorflow.keras.layers, losses, collections.Counter, Pandas, Numpy and sklearn. The TensorFlow GPU device [6] (i.e. /device: GPU:0) and TensorFlow version 2.1.0 is used for classification.

The following Figure 1 depicts the proposed sentiment classification process. An input sentence is segmented to 100 words as shown in the figure. These array words are embedded to the LSTM model though the drop-out layer, which is governed by “ReLU” and “Sigmoid” activators to avoid over-fitting. It is shown in the figure clearly that, the word[i] goes to the corresponding LSTM[i]. The outcome LSTM[i] goes to both the “Flatten” layer as well as to the succeeding LSTM[i+1] cells. Thereafter, “Dense” layers transform the classification to the “drop-out” layer

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

alternatively. The final classification with the sentiment extraction is captured in the "Final dense" layer.

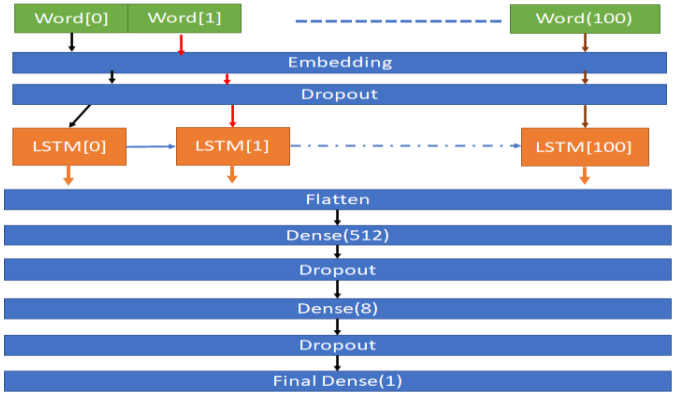

Figure 1: Flow of Sentiment Classification Process

1.3 PRE-PROCESSING

The following code removes the presence of any punctuation marks from the dataset present under consideration. It has the dictionary type variable "delete_dict1", which supervises the detection of any punctuation marks in the collection. A string table marker is used to translate the dictionary type variable into an indexed marker. Finally, the detected and modified table markers are split as per the following condition: the length of each finalized token should be greater than 2 irrespective of their case (upper case or lower case). The entire dataset is delivered as a .csv file.

```
def clean_DS_text(text_Raw ):
    delete_dict1 = {sp_character: '' for sp_character in string.punctuation}
    delete_dict1[' '] = ' '
    tbl = str.maketrans(delete_dict1)
    txt1 = txt.translate(tbl)
    textArr = txt1.split()
    txt2 = ' '.join([w for w in textArr if ( not w.isdigit() and  ( not w.isdigit() and len(w)>2))])
    return txt2.lower()
```


ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

From the initial pruned dataset (.csv file) [7], the following moderated dataset has been delivered. It is further processed by considering the following labels of dataset: Unnamed, rating, verified, reviewTime, reviewer ID, productID, reviewText, summary, and unixReviewTime. The entire dataset is grouped by increasing the order of ratings and the product ID. The following code performs the task above data categorization. The sample processed dataset is depicted in the following Figure 2.

```
rvw_data= pd.read_csv("..\\AmazonReviewsCellPhones\\CellPhonesRating.csv")
print(rvw_data.head(10))
print(len(rvw_data.groupby('productID')))
print(len(rvw_data.groupby('reviewerID')))
```

	Unnamed: 0	rating	verified	reviewTime	reviewerID	productID	\	reviewText	\	summary	unixReviewTime
0	0	5.0	True	08 4, 2014	A24E3SXTC62LJI	7508492919	0	Looks even better in person.	Can't stop won't stop looking at it		1407110400
1	1	5.0	True	02 12, 2014	A269FLZCB4GIPV	7508492919	1	When you don't want to spend		1	1392163200
2	2	3.0	True	02 8, 2014	AB6CHQWHZW4TV	7508492919	2	so the case came on time, i l		Its okay	1391817600
3	3	2.0	True	02 4, 2014	A1M117A53LEI8	7508492919	3	DON'T CARE FOR IT. GAVE IT /		CASE	1391472000
4	4	4.0	True	02 3, 2014	A272DUT8M88ZS8	7508492919	4	I liked it because it was cut		Cute!	1391385600
5	5	2.0	True	01 27, 2014	A1DW2L6XCC5TJS	7508492919	5	The product looked exactly li		Not so happy	1390780800
6	6	3.0	True	01 23, 2014	AQC61R4UST7UH	7508492919	6	I FINALLY got my case today.		It's cute!	1390435200
7	7	5.0	True	01 17, 2014	A31OVFL91BCKXG	7508492919	7	It is a very cute case. None		Cute case	1389916800
8	8	1.0	True	12 27, 2013	A1K0VLK6O5Z22M	7508492919	8	DO NOT BUY! this item is seri		WORST ITEM!	1388102400
9	9	4.0	True	12 16, 2013	A1K3BWU73YB44P	7508492919	9	I really love this case... yc		Pretty Cute!	1387152000

Figure 2: Sample Processed Amazon Review Dataset

A total of 938261 unique products and 47901 unique users are selected as per the pre-processed dataset amounting to a total of 986162 records. The distribution of records as per the rating in decreasing order is shown below (the maximum sentence length is limited to 4303 per int64 Type processing cycle):

5.0 rating – 555516

4.0 rating – 161434

3.0 rating – 90015

2.0 rating – 76692

1.0 rating – 54597

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

The general observation made above shows that, most of the ratings are above the 3.0 ratings, which is a preferred scenario for classification. An initial screening statistics is done with the help of the following code snippet and the generated box plot is captured in the Figure 3:

```
rating.set(style="whitegrid");rating.boxplot(x=rvw_data['Num_words']);
<AxesSubplot:xlabel='Num_words'>
```

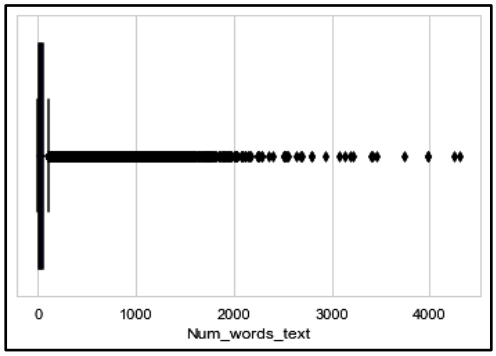

Figure 3: Box plot for the number of words in the dataset

The following code snippet classifies the above intermediary dataset into short and long reviews respectively. The code uses a mask_data as conditional connector, which checks and combines the review datasets, which have less than 100 words and greater than 20 words. This masked-up [8] dataset is then converted to a short review list. Python has a collection with the capability to store the heterogeneous data types. The same code has a mask_data conditional combiner altered for the condition to extract out sentence that has greater than 100 words. The same is categorized as the long review. The code is run over the review data amazon dataset and obtained the number of short reviews as 411313 and long reviews as 100239.

```
mask_data = (rvw_data['Num_words'] < 100) & (rvw_data['Num_words'] >=20)
set_shrt_rvws = rvw_data[mask_data]
print('No of Short reviews')
print(len(df_shrt_rvws))
mask_data = rvw_data['Num_words'] >= 100
set_long_rvws = rvw_data[mask_data]
```


ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

```
print('No of Long reviews')
print(len(df_long_reviews))
```

1.4 CREATING TRAINING AND TEST DATASETS

The following code snippet creates the training and test datasets required for the classification. This code uses a lambda function [9], which is an anonymous function that has proven strength across languages. The intermediary dataset obtained from short reviews are initially grouped by the “ProductID” tag. This grouped set is then transferred to the in-built function of Keras model [10] “filter”. A condition is set to restrict the filtering, whenever the length of a particular review is above 20. All these are fed as input to the sentiment extraction process based on the “ratings” tag. The filtered and applied dataset is then treated as training dataset, which is further used for the classification in this proposed model.

```
filtered_data = df_short_reviews.groupby('productID').filter(lambda x: len(x) >= 20)
filtered_data ['sentiment'] = filtered_data ['rating'].apply(get_sentiment)
train_data = filtered_data[['reviewText','sentiment']]
print(train_data['sentiment'].value_counts())
```

The following code snippet is responsible for the creation of the test dataset for classification. Here, the restriction set for the filter method is lesser than 100 as it is referring to the long reviews. Also, here the already masked reviews of the previous stages are used, subsequently the filter and applied methods are executed. The resultant dataset is treated as the test dataset for further classification.

```
Mask = review_data['Num_words_text'] < 100
df_short_reviews = review_data[mask]
filtered_data = df_short_reviews.groupby('productID').filter(lambda x: len(x) >= 10)
filtered_data ['sentiment'] = filtered_data ['rating'].apply(get_sentiment)
test_data = filtered_data[['reviewText','sentiment']]
```


ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

The outcome of the above code snippet is summarized in the following Table 1. From the table, it is evident that, three times the train data set is used for the test dataset, which is favorable for any meaningful classification. As per TensorFlow framework, these values are all tagged with int64 [11] data type by considering the 64 bit version of the platform.

Table 1: Train and test data distribution and count

S. No	Ratings	No. of words	Sentiment classification	Total words extracted
Train data				
1	5.0	119685	175910 (positive)	203891
2	4.0	36450		
3	3.0	19775		
4	2.0	15606	27981 (negative)	
5	1.0	12375		
Test data				
1	5.0	417691	592118 (positive)	686345
2	4.0	111226		
3	3.0	63201		
4	2.0	55746	94227 (negative	
5	1.0	38481		

From the generated dataset, the training and validation dataset has to be generated. This is accomplished by using the following code snippet. It is clearly seen from the code that it is two dimensional and a list is generated. At each of the record processing, the test size is kept as 0.5. This is to ensure that, 1:1 ration should be followed between the train and validation dataset [12], without any discrepancy in any of the two. The random state attribute of the train_test_split function aims to denote that non-linearity [13], which is maintained in the dataset pruning.

```
X_tr, X_valid, y_tr, y_valid = train_test_split(train_data['reviewText'].tolist(),\
                                train_data['sentiment'].tolist(),\
                                test_size=0.5,\
                                stratify = train_data['sentiment'].tolist(),\
                                random_state=0);
```

The outcome of the above code yields a train dataset of length 101945 and the validation dataset size as 101946. In both the cases, the distribution counter behaved with the following parametric values (1: 87955, 0: 13990).

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

After the above stage, the entire train and validation dataset is converted to text tokens by using the following text tokenizer. The property “unk” is used for the oov_token attribute [14]. The converted text token, which are in integer form that are made fit to the common tokens based on the statistical mean of each token.

```
tokenizer = Tokenizer(num_words=num_words,oov_token="unk")
tokenizer.fit_on_texts(X_train)
```

The outcome of the finalized training, validation and test dataset details are summarized in the following Figure 4.

```
======Train dataset ====
tf.Tensor(
[    4  1304   775    18  2423    41  4218  1450     2   178    41    69
   187     2   495 20463   173    47    70    17    23   391   422    57
    17   266     0     0     0     0     0     0     0     0     0     0
======Validation dataset ====
tf.Tensor(
[    2  1120  2842   775  1162     4   960    30    91    25  1040   451
     4   775     2  1393    34 44404    24  3623     3   437   144    34

======Test dataset ====
tf.Tensor(
[  78   83   91  787  646   11  211   33    6  600   57    2 3862   32
  400   48 5874   49 4917    7   45  307   10   32  197    9   68  104

shape=(100,), dtype=int32) tf.Tensor(0, shape=(), dtype=int32)
```

Figure 4: Summary of Train, Test and validation datasets

2. SENTIMENT MODEL USING LSTM

LSTM (Long-Short-Term-Memory) [15] is a neural network framework based on recurrent relations. LSTM has its promising applications in the field of deep learning. This framework uses the feedback connections unlike the feed forward connections of other classifiers. A sequence dataset such as Amazon review dataset are best fit with LSTM as state transducers are involved in each of the feedback state of the model. As the dataset under consideration is unsegmented, LSTM has its upper hold when compared to other deep learning algorithms.

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

The following Figure 5 illustrates the architecture and working of LSTM. The figure depicts the single sell state in LSTM classification. All the "C a.b" are the cell states. The "X_i" is the inputs and the corresponding outputs. This cell state has the input gate and forget gate [16] as a part of its architectures.

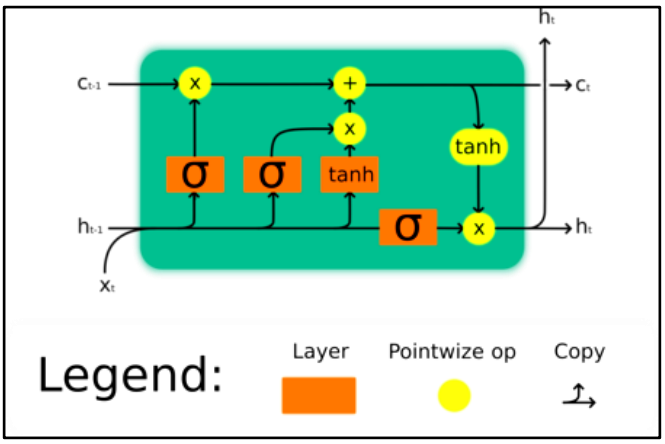

Figure 5: Cell wise architecture of LSTM Model

To work with the LSTM, one cells output is fed as input for the other cell. This sequence is fabricated further for this proposed model in Keras for the purpose of classification. Keras is an integrated part of TensorFlow2.1.0. Keras allows the entire platform to be with stack of layers.

2.1 KERAS MODEL FOR THE AMAZON REVIEW DATASET CLASSIFICATION

The following code snippet is the framework for the proposed model. The maximum number of features considered for the model classification is 50,000. The size of the vocabulary is also the same as that of the maximum features. The embedding layer has the dimension as 16. This denotes that every converted vector is in the 16 dimensional embedding space [17]. The embedding layer is added through the following function tf.keras.embedding(). The dense attribute has the activators as "ReLU" and "sigmoid". The entire compiled classification process is partitioned in to kernel regularizers and bias regularizers. These regularizers help in avoiding the over fitting as done in many of the classifiers. The drop out probability is set to 0.4, so as to restrict the vectors from discarded in the course of classification.

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

```
max_features = 50000
embedding_dim =16
model = tf.keras.Sequential()
model.add(tf.keras.layers.Embedding(max_features +1, embedding_dim,
input_length=sequence_length,\
                    embeddings_regularizer = regularizers.l2(0.005)))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.LSTM(embedding_dim,dropout=0.2, recurrent_dropout=0.2,
return_sequences=True,\ kernel_regularizer=regularizers.l2(0.005))
model.add(tf.keras.layers.Dense(512, activation='relu',\
                    kernel_regularizer=regularizers.l2(0.001),\
                    bias_regularizer=regularizers.l2(0.001),))
model.add(tf.keras.layers.Dropout(0.4)) model.add(tf.keras.layers.Dense(1,activation='sigmoid'))
model.summary()
```

The outcome of the LSTM model is summarized using the following code snippet and in Figure 6.

```
model.compile(loss=tf.keras.losses.BinaryCrossentropy(),optimizer=tf.keras.optimizers.Adam(1
        e-3),          metrics=[tf.keras.metrics.BinaryAccuracy()])
```

Model: "sequential_7"

Layer (type)	Output Shape	Param #
embedding_7 (Embedding)	(None, 100, 16)	800016
dropout_14 (Dropout)	(None, 100, 16)	0
lstm_7 (LSTM)	(None, 100, 16)	2112
flatten_7 (Flatten)	(None, 1600)	0
dense_21 (Dense)	(None, 512)	819712
dropout_15 (Dropout)	(None, 512)	0

Total params: 1,625,953
Trainable params: 1,625,953
Non-trainable params: 0

Figure 6: Parametric specifics of the LSTM model

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

2.2 TRAINING THE MODEL

The following code snippet demonstrates the training for model under consideration. The Epoch count of 10 is set. Epoch is nothing but neural network attribute which governs the proper fitting of training and test datasets. This Epoch count ensures the accurate prediction for any model. Higher the Epoch count makes the prediction more complex, yielding uneven results over varying data points. Lower the Epoch count makes the model more unstable. During every Epoch iteration, [18] the batch size of 1024 bytes is processed. The data points are observed in a shuffled order by ensuring that the model has no bias [19] over any other non-parametric vectors [20].

```
epochs = 10
history = model.fit(train_ds.shuffle(5000).batch(1024), epochs= epochs,
                    validation_data=valid_ds.batch(1024), verbose=1)
```

The following Figure 7 shows the accuracy of the model after 10 Epochs. The iterations 8, 9 and 10 are shown in the figure. At the end of 10th iteration the overall accuracy of 90.62% is attained.

```
Epoch 8/10
100/100 [==============================] - 14s 139ms/step - loss: 0.3124 - binary_accuracy: 0.8850 - val_loss: 0.2846 - val_bin
ary_accuracy: 0.9039
Epoch 9/10
100/100 [==============================] - 14s 139ms/step - loss: 0.3104 - binary_accuracy: 0.8851 - val_loss: 0.2824 - val_bin
ary_accuracy: 0.9049
Epoch 10/10
100/100 [==============================] - 14s 140ms/step - loss: 0.3127 - binary_accuracy: 0.8840 - val_loss: 0.2842 - val_bin
ary_accuracy: 0.9062
```

Figure 7: Prediction accuracy during the 8th, 9th and 10th Epoch iterations

2.3 MODEL ANALYSIS

A plot for measuring the loss during the text classification is made with the help of the following code. The plot is charted between “Number of Epochs” and “Loss Value”. Number of Epochs is kept in the x axis. The loss vale is kept in the y axis.

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

```
plt.plot(history.history['loss'], label=' training data');
plt.plot(history.history['val_loss'], label='validation data');
plt.ylabel('Loss value')
plt.xlabel('No. epoch')
plt.show()
```

The following observations are made for the train and test dataset performance using the Figure 8. A high degree of non-linearity in the progression exists in the initial values of the Epoch. The Epoch value of 4 initiates a smooth fitting for both the train and test dataset. Even then the alternate values of Epoch count iterations shows non-compliance with the linearity in the line. The training dataset experiences a high value as the volume is more compared to the test dataset.

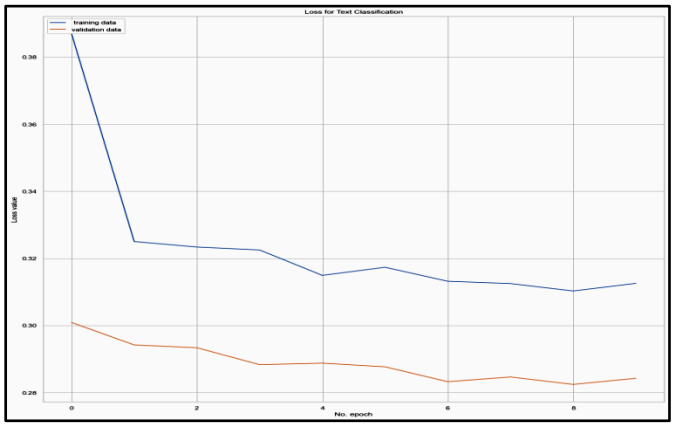

Figure 8: Correlations of Test and Train Datasets over the Epoch Iterations

The accuracy of the model is also plotted for the text classification using the following code snippet.

```
plt.plot(history.history['binary_accuracy'], label=' training data')
plt.plot(history.history['val_binary_accuracy'], label='validation data')
plt.title('Accuracy for Text Classification')
plt.ylabel('Accuracy value')
```


ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

```
plt.xlabel('No. epoch')
plt.legend(loc="upper left")
plt.show()
```

The Figure 9 illustrates the outcome of the model in terms of accuracy. The chart is plotted between “Number of Epochs” and “Accuracy value”. Both the training and test datasets are plotted.

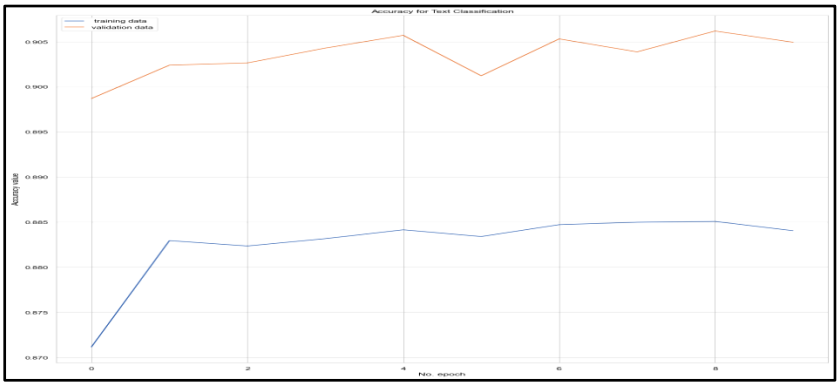

Figure 9: Text Classification Accuracy of Test and Train Datasets Volumes

Finally, a classification report is generated by using the following code snippet. The outcome of the above code is shown in the Figure X. The overall accuracy is 89%. The positive sentiments predictions are supported with the count of 94227 vectors. The negative sentiments predictions are supported with the count of 592118 vectors. This outcome has relevance with the accuracy chart that has been generated earlier in the Figure 10, where both the macro average and weighted average are having the same support count.

```
Labels = [0, 1]
print(classification_report(test_data['sentiment'].tolist(),test_data['pred_sentiment'].tolist(),labels
=labels))
```


ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

	precision	recall	f1-score	support
0	0.57	0.81	0.67	94227
1	0.97	0.90	0.93	592118
accuracy			0.89	686345
macro avg	0.77	0.86	0.80	686345
weighted avg	0.91	0.89	0.90	686345

Figure 10: Overall Accuracy and Support of the LSTM Model

CONCLUSION

In this research article, the Amazon review dataset is considered for analyzing the sentiments about mobile phone and accessories review of the users. The LSTM is applied in the Keras framework under the TensorFlow 2.1.0 platform. Python 3.6.0 is chosen as the preferred language of implementation, as it has many models for readily available libraries. A total of 100239 short (less than 20 words) and 411313 long reviews (greater than 20 words and less than 100 words) are considered for the classification. The maximum capability of the activators “ReLU” and “sigmoid” to avoid the over-fitting is maintained by restricting the feature size as 50,000 and embedding layers dimension as 16 to affect the extraction strategy. The “Epoch” iteration count of 10 is set to attain the maximum classification accuracy. The overall accuracy of 89% has been attained, while maintaining the balance between the macro and weighted average. The feedback connection mechanism of LSTM for every “Epoch” iteration gains the ability to provide the support count of 686345 words alignment. When classical models, which have prediction power similar to LSTM, are used to classify the same set of Amazon review dataset with similar volume, they misclassify even when the number of words exceeds by a little margin. This happens due to the inherent feed forward mechanism of the traditional models. Also, this is completely avoided and effectively handled by the LSTM model.

REFERENCES

[1] Chakrabarty, Navoneel, and Sanket Biswas. "Navo Minority Over-sampling Technique (NMOTe): A Consistent Performance Booster on Imbalanced Datasets." Journal of Electronics 2, no. 02 (2020): 96-136.

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

[2] Smys, S., and Jennifer S. Raj. "Analysis of Deep Learning Techniques for Early Detection of Depression on Social Media Network-A Comparative Study." Journal of trends in Computer Science and Smart technology (TCSST) 3, no. 01 (2021): 24-39.

[3] Haoxiang, Wang, and S. Smys. "Big Data Analysis and Perturbation using Data Mining Algorithm." Journal of Soft Computing Paradigm (JSCP) 3, no. 01 (2021): 19-28.

[4] Joe, Mr C. Vijesh, and Jennifer S. Raj. "Location-based Orientation Context Dependent Recommender System for Users." Journal of trends in Computer Science and Smart technology (TCSST) 3, no. 01 (2021): 14-23.

[5] Thilaka, B., Janaki Sivasankaran, and S. Udayabaskaran. "Optimal Time for Withdrawal of Voluntary Retirement Scheme with a Probability of Acceptance of Retirement Request." Journal of Information Technology 2, no. 04 (2020): 201-206.

[6] Siddique, Fathma, Shadman Sakib, and Md Abu Bakr Siddique. "Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers." 2019 5th International Conference on Advances in Electrical Engineering (ICAEE). IEEE, 2019.

[7] Moreno-Marcos, Pedro Manuel, et al. "Temporal analysis for dropout prediction using self-regulated learning strategies in self-paced MOOCs." Computers & Education 145 (2020): 103728.

[8] Li, Zhu, et al. "Kernel dependence regularizers and gaussian processes with applications to algorithmic fairness." arXiv preprint arXiv:1911.04322 (2019).

[9] Awan, Ammar Ahmad, et al. "HyPar-Flow: Exploiting MPI and Keras for Scalable Hybrid-Parallel DNN Training using TensorFlow." arXiv preprint arXiv:1911.05146 (2019).

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

[10] Adam, Edriss Eisa Babikir. "Deep Learning based NLP Techniques In Text to Speech Synthesis for Communication Recognition." Journal of Soft Computing Paradigm (JSCP) 2, no. 04 (2020): 209-215..

[11] Zou, Difan, et al. "Gradient descent optimizes over-parameterized deep ReLU networks." Machine Learning 109.3 (2020): 467-492.

[12] Goel, Priyanka, and S. Sivaprasad Kumar. "Certain class of starlike functions associated with modified sigmoid function." Bulletin of the Malaysian Mathematical Sciences Society 43, no. 1 (2020): 957-991.

[13] Chakraborty, Rupak, Rama Sushil, and M. L. Garg. "An improved PSO-based multilevel image segmentation technique using minimum cross-entropy thresholding." Arabian Journal for Science and Engineering 44.4 (2019): 3005-3020.

[14] Pooja.C, Thivaharan.s, "Workload based Cluster Auto Scaler using Kuberbet Monitors", International Journal Compliance Engineering Journal (IJCENG), 2021, Vol.12, Issue.6, pp. 40-47, ISSN:0898-3577, DOI:16.10089.CEJ.2021.V12I6.285311.36007.

[15] S. Thivaharan., G. Srivatsun. and S. Sarathambekai., "A Survey on Python Libraries Used for Social Media Content Scraping," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp. 361-366, doi: 10.1109/ICOSEC49089.2020.9215357.

[16] Lnenicka, Martin, and Jitka Komarkova. "Big and open linked data analytics ecosystem: Theoretical background and essential elements." Government Information Quarterly 36.1 (2019): 129-144.

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

[17] Dube, Thando, Rene Van Eck, and Tranos Zuva. "Review of Technology Adoption Models and Theories to Measure Readiness and Acceptable Use of Technology in a Business Organization." Journal of Information Technology 2, no. 04 (2020): 207-212.

[18] Smilkov, Daniel, Nikhil Thorat, Yannick Assogba, Ann Yuan, Nick Kreeger, Ping Yu, Kangyi Zhang et al. "Tensorflow. js: Machine learning for the web and beyond." arXiv preprint arXiv:1901.05350 (2019).

[19] Manoharan, Samuel. "Embedded Imaging System Based Behavior Analysis of Dairy Cow." Journal of Electronics 2, no. 02 (2020): 148-154.

[20] Salman Taherizadeh., VladoStankovski., “Auto-scaling Applications in Edge Computing: Taxonomy and Challenges” Conference: International Conference on Big Data and Internet of Thing (BDIOT2017) - ACM, At London, United Kingdom

AUTHORS BIOGRAPHY

[1]Mr. S. Thivaharan did his B.Tech (IT) from Thiagarajar College of Engineering. He completed his M.Tech in Networking from Amrita School of Engineering. He is presently pursuing PhD in Anna University, Chennai. He worked as a system Engineer in TATA Consultancy Services Ltd, Chennai for 3 years. Then he switched to teaching profession. He is presently working as Assistant Professor (Selection Grade) in PSG Institute of Technology and Applied Research, Coimbatore. He is having a total of 12 years of teaching experience.

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021

Journal of Artificial Intelligence and Capsule Networks (2021)
Vol.03/ No.02
Pages: 72-89
http://irojournals.com/aicn/
DOI: https://doi.org/10.36548/jaicn.2021.2.001

[2]Dr. G. Srivatsun did his BE (ECE) from Bharathiyar University. He Completed his ME (Wireless technologies) from Anna University and PhD in the area of RF antennas from Anna University, Chennai. After a year of service from affiliated institution, he joined PSG college of Technology. He has won the AICTE - Career award for young teachers during the year 2013 for his research in antennas, through which he has traveled to USA. he has published various national and international journals. He has also served as faculty advisor for student affairs, coordinated the dean functioning and research activities. He is the nodal officer for All India Survey for Higher Education (MHRD).

ISSN: 2582-2012 (online)
Submitted: 27.04.2021
Revised: 21.05.2021
Accepted: 7.06.2021
Published: 15.06.2021